TEXTURE PROCESSING APPARATUS AND METHOD

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit of Korean Patent Application No. 10-2015-0107516, filed on Jul. 29, 2015, in the Korean Intellectual Property Office, the entire disclosure of which is incorporated herein in by reference for all purposes.

BACKGROUND

[0002] 1. Field

[0003] The following description relates to texture processing apparatuses and methods.

[0004] 2. Description of Related Art

[0005] A graphics processing unit (GPU) includes a texture processor to reduce an amount of calculation and increase rendering speed during a pixel shading process of a three-dimensional (3D) graphics rendering process.

[0006] The texture processor performs a process of generating a texture required for texturing. Texturing is a process of adding a prepared image to an object formed in a 3D space, and is one of operations included in 3D graphics rendering for reducing the amount of calculation performed. The prepared image is referred to as a texture. The texture may be compressed and stored beforehand in an external memory of the texture processor.

[0007] The texture processor transmits a texture required by a shader core upon receiving, from the external memory, a compressed texel block obtained by compressing texels constituting a texture in predetermined block units, and processing the compressed texel block.

SUMMARY

[0008] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0009] According to one general aspect, a texture processor includes a texture cache configured to store textures; a controller configured to determine a texture address corresponding to a requested texture and read a texture, among the stored textures, corresponding to the texture address from the texture cache; a format converter configured to convert a format of the read texture into another format, based on a degree of texture precision required by a graphics processing unit (GPU); and a texture filter configured to perform texture filtering using the texture having its format converted into the another format.

[0010] The format converter may be configured to determine an output format of the texture based on the degree of texture precision required by the GPU, and output the read texture in the determined output format.

[0011] The format converter may be configured to determine the degree of texture precision required by the GPU based on at least one of a texture filtering mode and a type of an application programming interface (API) used in the GPU, and determine whether the format of the read texture is to be converted, based on a result of the determining of the degree of texture precision.

[0012] The format converter may be configured to convert the format of the read texture into the another format when the texture filtering mode is not a preset mode that requires that the format of the read texture be maintained and the type of the API is not a preset type that requires a high degree of precision.

[0013] The format converter may be configured to determine the degree of texture precision required by the GPU by further taking into account a level of rendering that the GPU is requested to perform.

[0014] The format converter may be configured to convert the format of the read texture into the another format when the texture filtering mode is not a preset mode that requires that the format of the read texture be maintained, the type of the API is not a preset type that requires a high degree of precision, and the level of rendering that the GPU is requested to perform is less than or equal to a predetermined level.

[0015] The level of rendering that the GPU is requested to perform may be determined by a user through an application linked to the GPU or determined by the GPU based on hardware resources of the GPU.

[0016] The texture filter may be configured to perform texture filtering by driving a filter corresponding to the converted format.

[0017] The texture processor may further include a decompressor configured to decompress the read texture. The format converter may be configured to convert a format of the decompressed texture into another format, based on the degree of precision of the texture required by the GPU.

[0018] The texture processing unit may further include a decompressor configured to decompress the read texture having its format converted into the another format. The texture filter may be configured to perform texture filtering using the decompressed texture.

[0019] According to another general aspect, a texture processing method includes: determining a texture address corresponding to a requested texture; reading the texture corresponding to the texture address from a texture cache; converting a format of the read texture into another format, based on a degree of texture precision required by a graphics processing unit (GPU); and performing texture filtering using the read texture having its format converted into the another format.

[0020] The converting of the format of the read texture may include: determining an output format of the read texture based on the degree of texture precision required by the GPU; and outputting the read texture in the determined output format.

[0021] The determining of the output format of the texture may include determining the degree of texture precision required by the GPU based on at least one of a texture filtering mode and a type of an application programming interface (API) used in the GPU; and determining whether the format of the read texture is to be converted, based on a result of determining the degree of texture precision.

[0022] The determining of whether the format of the texture is to be converted may include converting the format of the read texture into the another format when the texture filtering mode is not a preset mode that requires that the format of the read texture be maintained and the type of the API is not a preset type that requires a high degree of precision.